

2 用户界面介绍

2.1 参数窗口

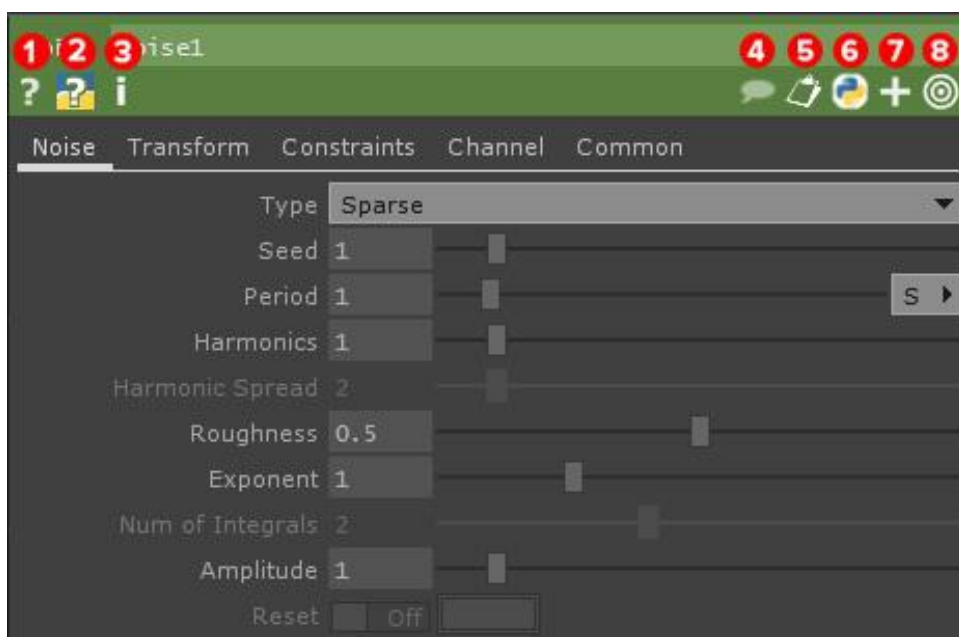
在参数窗口可以访问所有 OP 的参数。

有两种方法打开它。

一种是用“P”键，这样会打开一个停靠在窗体右上角的固定窗口。它会显示任何你当下选择的 OP 的参数。

另一种是直接右键 OP，选择 Parameter(参数)，这样会打开一个浮动窗口。这种方法的区别是，此时如果你选了别的 OP，这个窗口的参数还是之前那个 OP 的参数。

想要同时操作多个 OP 的参数时，第二种方法会很有效。虽然每个 OP 的参数设置都不相同，但它们都用相同的参数选项。下面是个某个 OP 选项的参数。



从左往右，这些参数分别是：

1. 元件帮助：在浏览器新窗口中打开 OP 的 WIKI 帮助页面。
2. 元件 Python 帮助：在浏览器新窗口中打开 OP 的 Python 帮助页面。
3. OP 信息对话框：显示 OP 程序的信息，效果等同于中键点击 OP
4. 备注：显示和编辑 OP 的备注
5. 复制参数：显示通过邮件点击菜单复制的参数
6. 语言：选择用 Python 还是 tscript 作为 OP 的脚本语言
7. 展开/合并参数：展开或合并显示这个 OP 的全部参数
8. 非默认参数：只显示被改动过，非默认状态的参数。

2.2 参数

有多种方式可以设置参数。根据情况，有些参数需要是静态值，另一些参数需要随别的变量或输入而变动。每个参数有三种类型。这三种类型完全不同，每种都决定了参数的运行方式。他们是：

常数类型

表达式类型

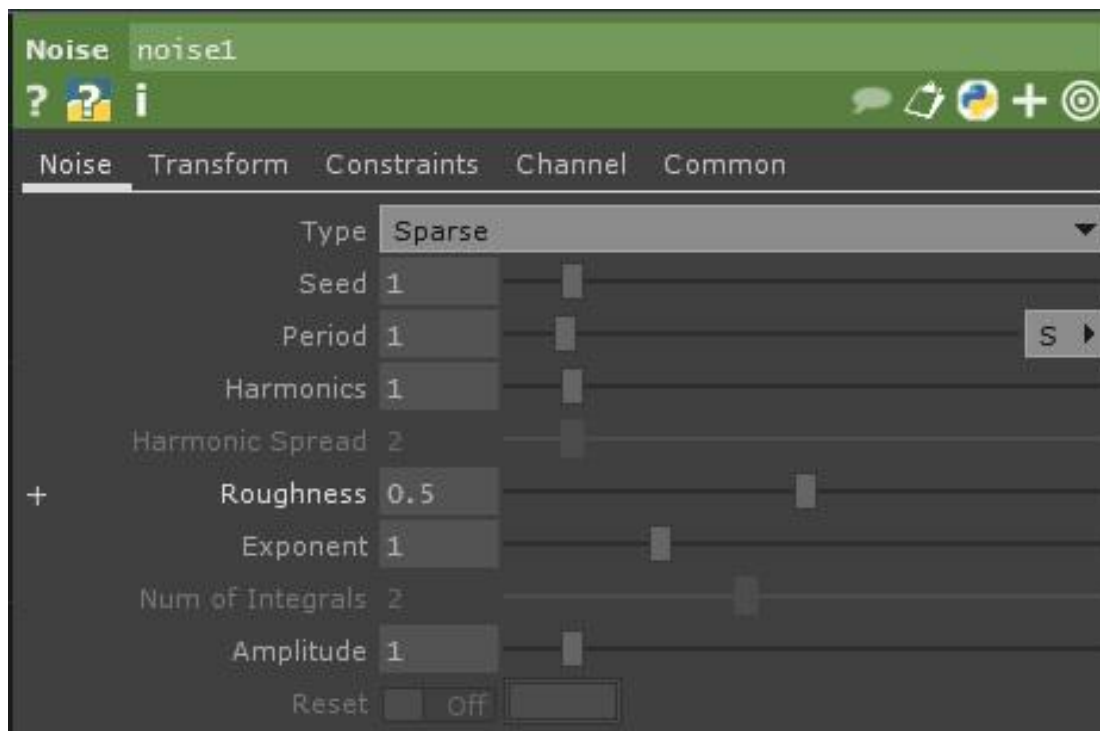
输出类型

大多参数的默认类型是常数型，它的数值区域显示为灰色。

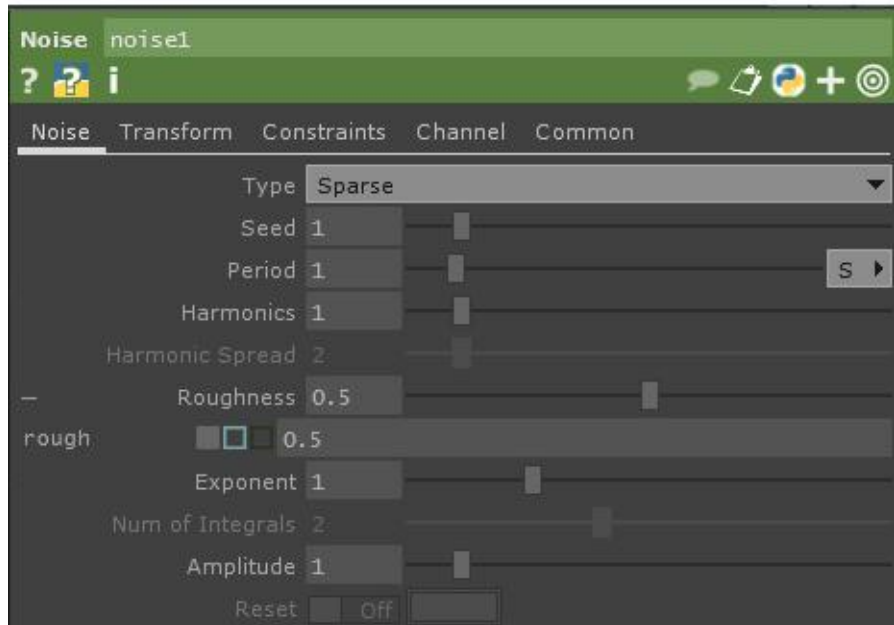
表达式类型用于 Python, tscript, 数学运算或脚本，颜色为深灰色和亮蓝色组合。

输出类型直接引用 channel 的取值，它的表现颜色为亮绿色。

每一个 OP 参数都可以在这三种类型之间切换，只要把鼠标悬停在参数名称上，参数名称附近会出现一个加号，就像下图：



点击加号，它会展开显示更多信息和选项，就像下面：



当这个参数被点击后，出现三个可变的元素。

第一个是这个参数脚本的名字。

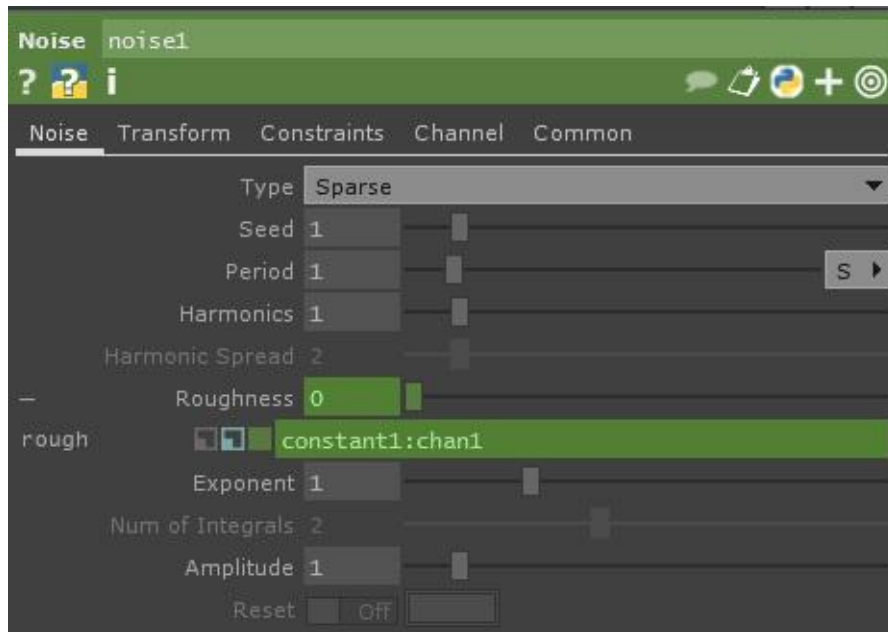
在 TD 的任何一种脚本语言中，任何时候引用参数，都需要脚本名称。这上面那个例子里，Noise CHOP 的 Roughness 参数的脚本名称是 'rough'。

在 Python 中设置 Roughness 的值为 1 的语句如下：

```
op('noise').par.rough = 1
```

第二个元素是三个不同颜色的方块。他们代表参数的不同类型，就像上面提到过的。默认为灰色的常数类型，点击蓝色方框，可以变为表达式类型。点击后，右边的参数值也会根据类型不同，变为对应颜色。

要是想变成输出类型，需要把某个 CHOP 元件的 channel 拖到参数上，然后输出类型的小方框会被自动选上，后面的背景色也变成了绿色。



第三个元素是数值区域。它显示所选类型的一些信息。

常数类型下，它显示当前值，并且可以被编辑和输入。

表达式类型下，它显示对应的 Python 脚本。这些表达式可以被输入和编辑。

输出类型下，

数值区域被一个冒号分为两部分。冒号前是来源元件的名字，冒号后是那个来源元件中的参数名称。因为这些值会被其他 OP 应用，所以在输出类型下，参数不能被编辑。

2.3 运动控制

TD 运动控制栏的功能和很多软件是一样的。咱们从左往右，把这些按钮过一遍。



1. 把时间轴拖到第一帧
2. 暂停时间轴
3. 反向播放
4. 正向播放
5. 后退一帧
6. 向前一帧
7. 在选定范围循环播放
8. 在选定范围播放一次，并停在最后一帧

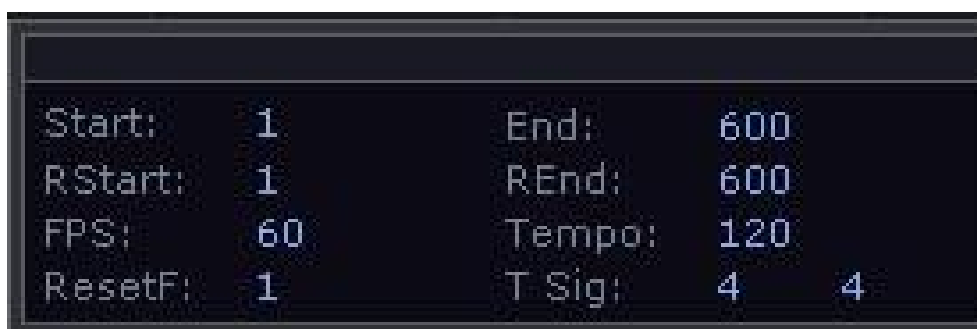
最常用的功能是开始和暂停。用空格键实现。

2.4 时间轴设置

除非有视频或动画被锁定到时间轴上了，否则一般来说，不用经常操作时间轴的设置。时间轴设置在屏幕左下角，重点是要知道，这里可以改变工程运行的帧速度或节奏。帧速度决定了项目渲染每一帧的速度，默认是 60 帧/秒。也就是说工程会尽量保证每秒渲染 60 帧画面。

节奏决定了项目每分钟的节拍数，Beat CHOP 这东西会用到它。

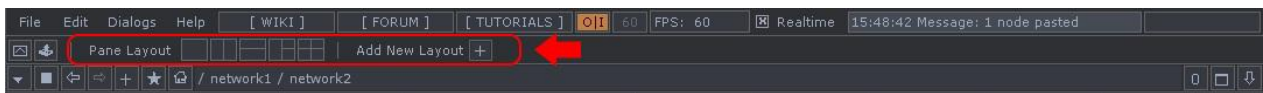
时间轴设置常用在媒体或者动画需要被锁定在一个一致的时间轴上的情况。帧控制包括 'Start' 和 'End' ,用来控制时间轴的第一帧和最后一帧。同样，'RStart' and 'REnd'则是用来控制时间轴循环播放的开始点和结束点。通过这些设置，既能创建一个横跨整个时间轴的 4000 帧动画，也能实现循环播放时间轴的一小部分。

A screenshot of the Time Axis settings panel in TouchDesigner. The panel has a dark background with light blue text. It contains several parameters arranged in two columns. The first column includes 'Start:', 'RStart:', 'FPS:', and 'ResetF:'. The second column includes 'End:', 'REnd:', 'Tempo:', and 'T Sig:'. The values for these parameters are: Start: 1, RStart: 1, FPS: 60, ResetF: 1, End: 600, REnd: 600, Tempo: 120, and T Sig: 4 4.

Start:	1	End:	600
RStart:	1	REnd:	600
FPS:	60	Tempo:	120
ResetF:	1	T Sig:	4 4

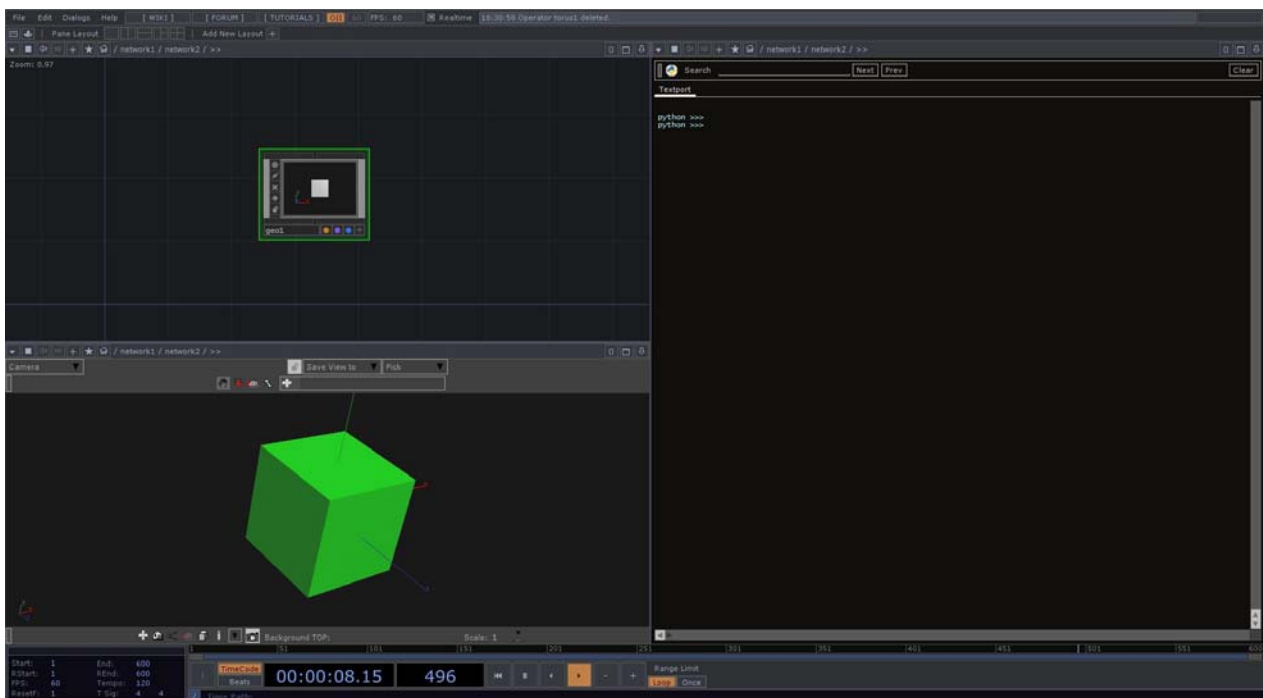
2.5 分屏显示

在网络之间来回移动和跳转的时候，经常用分屏可以节约很多时间。你要是穿过三层网络来改变一个参数，哪怕只是跳回去看一下变更，都会很浪费时间。而用分屏功能可以把窗口分割成任意块，这样就不用跳来跳去了，每个面本都能保存以供将来使用。



上图表示了默认可用的几种预置的分屏类型。预置提供了几种快捷的标准配置，包括左右分，上下分，分成三块以及四块。保存设置也很简单，点击'Add New Layout +'按钮，输入名字，点 OK。保存布局不止保存分屏的位置和尺寸，也会保存每个分屏的类型。

分屏面板可以用来显示独一无二的内容，不论是对话框，网络，还是视角。可以灵活的混合和比较组合的视角或网络编辑器。下图中，左上角的面板是个网络编辑器，右手边是个 Textport，左下角是个几何视角。同样，保存布局不止保存了面板的排列，也保存了面板类型。当搭建一个有很多元素的项目时，这一点很有用。当需要上面这些设置之间跳转或是进入一个小的网络编辑器中时，会节约很多时间。



关于面板分屏的快捷键有这些：

1. Alt + [: 从鼠标处垂直切分面板
2. Alt +] : 从鼠标处水平切分面板
3. Alt + Z : 关闭鼠标所在面板

2.6 资源查看器（组件库）

资源查看器可以被看作是一个组件库。资源查看器用来汇总'.tox'文件（或者 TD 组件文件）。这些文件包含有一个 OP 组件，而这种 OP 组件保存着一个网格或者 OP 元件。这意味着，一些常用的 OP，网格，组件，脚本，可以被创建进一个单独的 OP 组件，存为'.tox'，并可随时调用。

打开资源查看器，然后浏览这些已经存在的可用的.tox 文件。空项目的资源查看器默认是打开的，停靠在窗体左侧。如果要使它成为一个悬浮窗体，用'Alt + L'。让我们试试其中一个已经预先构建的组件。

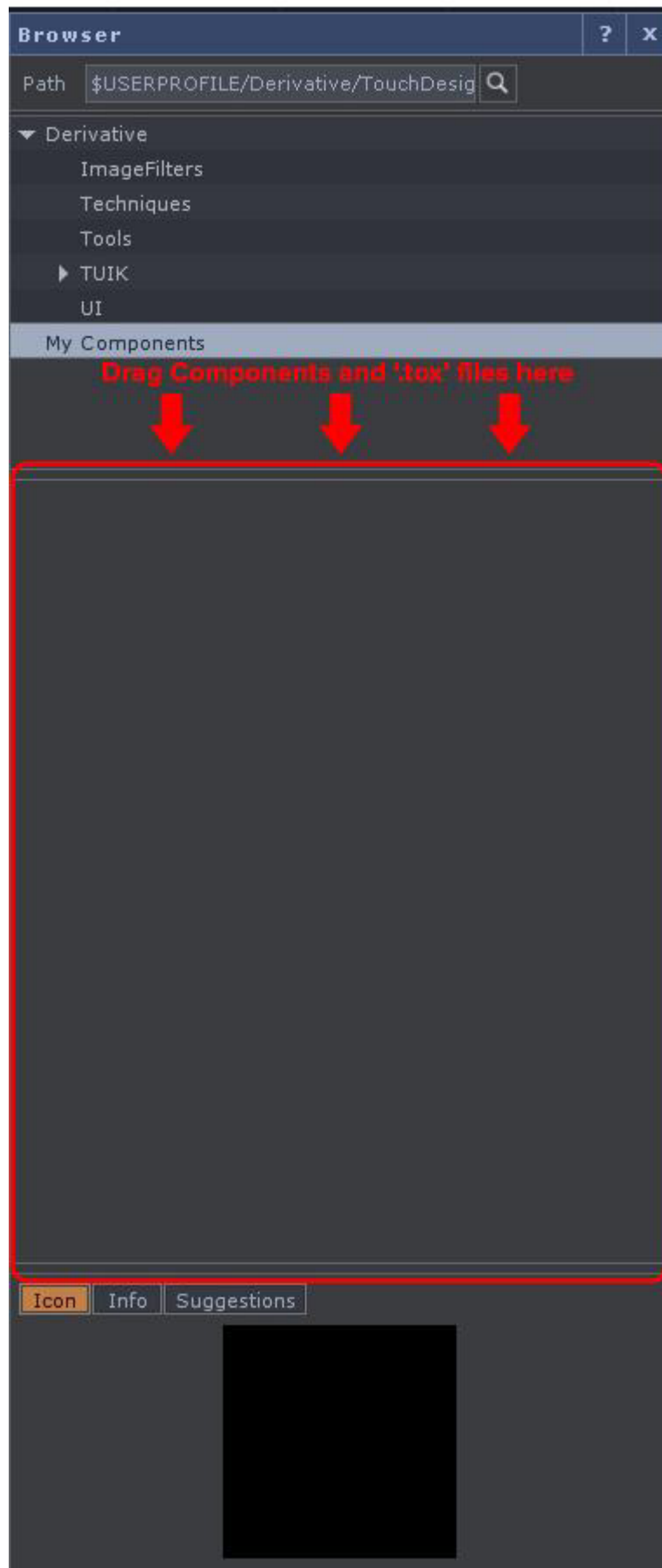
在'Derivative'区域，找到'Tools'，然后把'BlendModes'组件拖放到一个新项目中。观察'BlendModes'组件的 UI，很显然，它里面有很多东西在运行。在深入了解前，先连接两个输入，试着操作下'BlendModes'组件。你可以通过导入两个不同的图层来观察不同的叠加效果。这个工具很实用，而全部操作只是在资源查看器中的一个拖放动作。

本书的目标之一就是创建一些工具，使它们可以添加到资源查看器，以便可以被经常使用。有两种方法可以将组件添加到资源查看器中。

第一种是拖放。在查看器顶部选择'My Components'，然后拖动网格中的任意组件并放进资源查看器的下面，它就会被添加进'My Components'库。

第二种方式是从 Windows Explorer 中拖动一个已经保存的'.tox'文件，把它放到前面提到的区域。下图准确描述了组件应该被放到哪里。

Introduction To TouchDesigner



2.7 搜索对话框

当项目变得越来越复杂，充斥着无数的 Python 脚本或是嵌套的网络，搜索对话框会变得超乎想象的有用。它可以用于很多区域，查找很多东西，而且根据需要，可以选择模糊搜索或者精确搜索。按 F3 或在顶部的'Edit'菜单中打开搜索。

常规搜索不仅能查找 OP 元件你还能查找 Python 代码。通常，Python 用来改变 OP 参数值。有时候，在一大堆 Python 脚本里，常常不易找到一些特定行的代码。

比如利用下面这行代码可以找到所有与名为 transform 的元件的参数相关的代码信息。

```
op('transform').par
```

从结果中筛选代码比从网络中笼统寻找要容易的多。

“高级”搜索可以搜索任何名称、OP 类型、注释、标志等的组合。

当处理老项目时，常常要花不少时间来回忆当时构建的那个，复杂逻辑网络的内在工作流程。这时就用模糊的名称和类型搜索，会节约很多时间。比如，在一个高度嵌套化的系统里，在深层中的某个地方，可能有个影片，名字中有'movie'几个字母，是 TOP 类的元件，这些 OP 类型信息，和名字的片段，可以用来进行一次精准的搜索。

当获得搜索结果，每个结果都可以在点击后，在一个新的浮动窗口中打开。这些结果可以在这里被快速预览，简单编辑，并不会影像当前的工作区域。

2.8 实时渲染



实时渲染能合理地分配资源，在 GPU 不足时也可以进行编辑。当它被激活（默认激活），TD 会优先使用真实时间。举个简单例子，如果一个影片长 30 秒，不论发生什么，TD 会让他播放 30s。如果因此导致帧速度被降低，TD 会试着用虚拟时间。这种模式在实时装置和高性能要求的工作中经常用到。

当实时渲染被关掉，TD 将优先渲染真实时间。在上面提到的例子中，如果关掉实时渲染，TD 会按它需要来运算渲染每一帧，脱离真实世界的时间来显示每一帧。这个模式在导入复杂的动画或 3d 模型时很有用。想想下，这个模式跟 Adobe After Effects 渲染时的实时画面比较像。

2.9 有用的快捷方式

当鼠标悬停在网络上方，下面的快捷键可以使用：

'P' 打开/并关闭选定 OP 的参数窗口

'O'在面板左下角打开和关闭当前网络的概览。

'C'打开和关闭资源查看器。这可以为选定的 OP 添加一个彩色的轮廓，以方便识别

'A'允许与 OP 在状态窗口和编辑窗口之间进行切换

'B'忽略或取消忽略选择的 OP

'H'执行总览网络的动作，实现在屏幕中看到当前网络的所有 OP

'Shift + H' 在屏幕中查看所有选择的 OP

'R' 切换 OP 的渲染标志（如果有）

'D' 切换 OP 的显示标志（如果有）

'Control + C' 复制选择的 OP

'Control + V' 粘贴复制的 OP

'Control + Shift + V' 在鼠标处粘贴复制的 OP